

# Community detection in directed acyclic graphs<sup>★</sup>

Leo Speidel<sup>1,2</sup>, Taro Takaguchi<sup>2,3</sup>, and Naoki Masuda<sup>4,a</sup>

<sup>1</sup> Department of Mathematical Informatics, The University of Tokyo, Hongo 7-3-1, Bunkyo-ku, Tokyo 113-8656, Japan

<sup>2</sup> JST, ERATO, Kawarabayashi Large Graph Project, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan

<sup>3</sup> National Institute of Informatics, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan

<sup>4</sup> Department of Engineering Mathematics, University of Bristol, Woodland Road, Clifton, Bristol BS8 1UB, UK

Received 20 March 2015 / Received in final form 27 June 2015

Published online 10 August 2015

© The Author(s) 2015. This article is published with open access at [Springerlink.com](http://Springerlink.com)

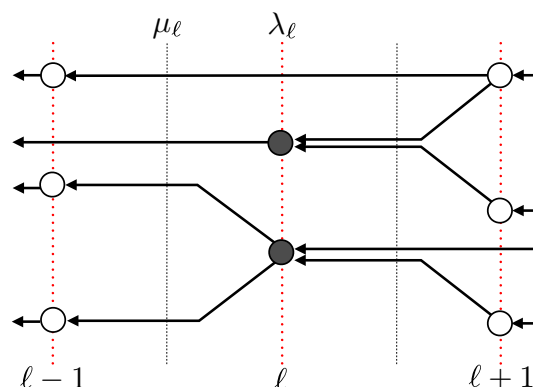
**Abstract.** Some temporal networks, most notably citation networks, are naturally represented as directed acyclic graphs (DAGs). To detect communities in DAGs, we propose a modularity for DAGs by defining an appropriate null model (i.e., randomized network) respecting the order of nodes. We implement a spectral method to approximately maximize the proposed modularity measure and test the method on citation networks and other DAGs. We find that the attained values of the modularity for DAGs are similar for partitions that we obtain by maximizing the proposed modularity (designed for DAGs), the modularity for undirected networks and that for general directed networks. In other words, if we neglect the order imposed on nodes (and the direction of links) in a given DAG and maximize the conventional modularity measure, the obtained partition is close to the optimal one in the sense of the modularity for DAGs.

## 1 Introduction

Temporality and community structure are two common features present in various types of network data. Temporality of networks refers to nodes and links that vary over time. For example, a friendship link between a given pair of individuals is not always used even if they are close friends of each other. The link would be only occasionally active as the two individuals meet and then separate. Temporally varying networks are collectively called temporal networks [1]. Community structure posits that nodes or links in networks can be classified into groups, called communities [2]. Typically, a community is defined such that links are dense within a community and relatively sparse across different communities. Many networks in different domains have community structure.

The two features can be naturally combined into community detection in temporal networks, and several algorithms have been proposed to this aim. Examples include cost minimization when temporal non-smoothness is a part of the cost function [3,4], optimization under temporal smoothness constraints [5], methods based on the Potts model [6,7], clique percolation [8], decomposition of adjacency tensors [9], generalization of modularity for adjacency tensors [10–12], link clustering [13], and the minimum description length principle [14].

Related to temporal networks is the concept of directed acyclic graph (DAG). DAGs are directed networks



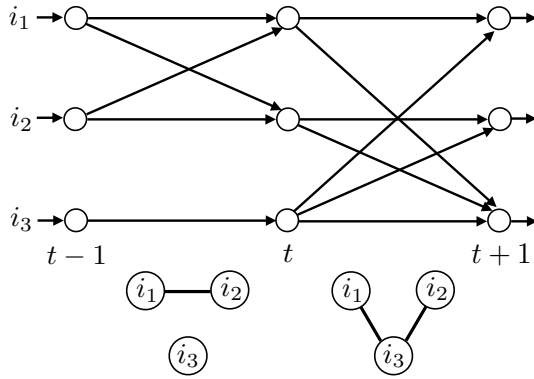
**Fig. 1.** Schematic of a DAG magnified around layer  $\ell$ . The indices  $\ell - 1, \ell, \ell + 1$  represent three subsequent layers from lower to upper. The filled circles represent the nodes in layer  $\ell$ . The quantities  $\mu_\ell$  and  $\lambda_\ell$  are given by the number of links passing through the corresponding vertical lines.

without directed cycles. In DAGs, nodes can be positioned within a layer structure such that links only emanate from a node in a higher layer to a node in a lower layer (Fig. 1). DAGs have been common as a tool for statistical inference for decades [15,16]. Equally importantly, we find various instances of DAGs in the real world such as some food webs [17], some dominance hierarchy networks [18,19], citation networks [20–23], family trees [24], and phylogenetic networks [25].

Temporal networks can be mapped to DAGs in at least two ways. First, citation networks, a type of temporal

<sup>★</sup> Contribution to the Topical Issue “Temporal Network Theory and Applications”, edited by Petter Holme.

<sup>a</sup> e-mail: [naoki.masuda@bristol.ac.uk](mailto:naoki.masuda@bristol.ac.uk)



**Fig. 2.** Schematic of the DAG representation of a temporal network, in which the link configurations between the same set of nodes change over time. The lower row shows snapshots of the network taken at times in  $[t-1, t)$  and  $[t, t+1)$ , respectively.

network, can be naturally mapped to DAGs. A citation network is a directed network in which a node represents an article such as a scientific paper, patent, or court decision, depending on the network, and a link is directed from the citing to the cited nodes [26]. It is temporal in the sense that it grows over time due to the addition of new nodes and links [1]. In principle, the links are directed backwards in time because newer nodes can cite older nodes but not vice versa, which makes the network a DAG. There may be links contradicting the arrow of time in real data sets, such as mutual citations [26], which would violate the definition of a DAG. However, these links are relatively few (see Sect. 5.1 for exemplar numbers).

Second, a family of temporal networks can be mapped to DAGs, as schematically shown in Figure 2. Consider a sequence of adjacency matrices indexed by time, i.e.,  $(A^{(t)})_{t=1,\dots,T}$ , where  $t$  is discrete time,  $(A^{(t)})_{ij} = 1$  if  $i$  and  $j$  are adjacent to each other in the  $t$ th snapshot, and  $(A^{(t)})_{ij} = 0$  otherwise. By definition,  $(A^{(t)})_{ij} = 1$  implies that  $i$  and  $j$  contacted each other at some (continuous) time contained in the time interval  $[t, t+1)$ . Such a representation of temporal networks as sequences of matrices can be induced by the temporal resolution of the recording device or by the aggregation of continuous-time temporal network processes over a finite time window to create a snapshot [1]. In Figure 2, each node  $i$  is duplicated  $T$  times, and each duplicated node is labelled  $(i, t)$ , where  $1 \leq t \leq T$ . Therefore, there are  $NT$  nodes in total in the representation shown in Figure 2, where  $N$  is the number of nodes in the original temporal network. We draw a link from  $(i, t)$  to  $(i, t+1)$  for every node  $i$  ( $1 \leq i \leq N$ ) and  $1 \leq t < T$ . We also draw a link between two nodes in subsequent layers if they are connected in the corresponding time interval. In other words, we draw a link from  $(j, t)$  to  $(i, t+1)$  if  $(A^{(t)})_{ij} = 1$ . In this way, a temporal network given as a sequence of adjacency matrices is uniquely mapped to a DAG in which links only span between two subsequent layers. This type of representation and its variants have been used in the analysis of temporal networks [27–29].

Community detection methods for temporal networks mentioned earlier in the present section are designed for the latter type of representation. However, to the best of our knowledge, community detection methods explicitly designed for the former type of temporal networks, or more generally DAGs, have not been proposed so far. Community structure in the former type of temporal networks has been studied using methods such as agglomerative hierarchical clustering [30], conventional undirected and directed modularity [21,31], and the Infomap method [32]. These detection methods are designed for static networks. They neither incorporate the acyclic nature of citation networks nor temporal information such as the publication dates of articles. In these studies, temporal dynamics were analyzed once communities were obtained by the static methods. Therefore, if we apply existing methods, we have to map a temporal network of the former type to a static network by discarding the temporal information or to a snapshot representation. Both types of mapping seem to be suboptimal given the natural representation of the original network as a DAG.

In the present study, we develop a community detection method which exploits the intrinsic temporality in the former type of temporal networks. Technically, we propose a community detection method for general DAGs. We do so by developing a modularity measure and a maximization method for DAGs. A key to this development is the observation that the choice of a null model network characterizes communities to be detected. Null models randomize links while preserving some properties of the original network. Communities obtained by modularity maximization therefore are structures that are statistically surprising relative to the null model. Depending on the class of networks, specific null models have been proposed. Examples include networks without multilinks and selfloops [33], weighted networks [34], directed networks [35,36], multi-partite networks [37–39], spatial networks [40], networks with a similarity measure imposed between nodes [41], networks directly formed by correlation matrices [42], and multilayer networks [10–12] in which temporal networks are included as a special case. If we use an inappropriate null model to detect communities in a DAG, we may obtain a suboptimal result. In particular, all aforementioned methods do not respect the directed layer structure inherent in a DAG. With such a method, a modularity value might be large simply because a DAG is surprising as compared to a non-DAG null model. However, we are interested in modular structure that a given DAG may have whereas reference DAGs do not. Therefore, in the present study we propose a modularity maximization method that uses a null model for DAGs.

## 2 DAG

Let us denote a directed network by  $G = (V, E)$  with  $|V| = N$  nodes and  $|E| = M$  directed links. We assume that  $G$  is a DAG. We also assume that  $G$  has  $L$  layers, that node  $i$  ( $1 \leq i \leq N$ ) belongs to layer  $\ell_i$  ( $1 \leq \ell_i \leq L$ ), and that any directed link from  $j$  to  $i$  satisfies  $\ell_j > \ell_i$  (Fig. 1).

In words, any link emanates from a node in the upper layer (i.e., larger layer number) to a node in the lower layer (i.e., smaller layer number). No directed link exists between nodes in the same layer. A layer may include multiple nodes. For example, in Figure 1, layers  $\ell - 1$ ,  $\ell$ , and  $\ell + 1$  contain three, two, and three nodes, respectively. No order is imposed between nodes in the same layer.

Our definition is motivated by empirical observations that it is often more appropriate not to impose an order in subsets of nodes. For example, in citation networks, articles published on the same date cannot cite each other unless the two papers coordinate their citations [21–23,26]. In dominance hierarchy networks of animals, it may be difficult to rank individual animals (i.e., nodes) if they have similar physical strengths and no prior interaction, which would be the case in large colonies.

We can always give such a layering to a given DAG although layering is not unique in general [43]. In the remainder of the present article, we simply refer to a DAG supplied with a proper layering  $\{\ell_1, \dots, \ell_N\}$  as a DAG.

### 3 Modularity

Communities detected through modularity maximization depend on the null network model that serves as a baseline for defining the modularity. In this section, we first briefly review modularity measures for undirected and directed networks allowing for cycles. Then, we formulate modularity for DAGs by proposing a new null model tailored to the case of DAGs.

#### 3.1 Modularity for undirected networks

The modularity for undirected networks is defined by:

$$Q^{\text{und}} \equiv \frac{1}{2M} \sum_{i,j=1}^N \left( A_{ij} - \frac{k_i k_j}{2M} \right) \delta_{c_i, c_j}, \quad (1)$$

where  $A_{ij}$  is the  $(i, j)$  element of the adjacency matrix, and  $k_i = \sum_{j=1}^N A_{ij}$  is the degree of node  $i$  [2,44]. The adjacency matrix is defined by  $A_{ij} = 1$  if nodes  $i$  and  $j$  are adjacent by a link and  $A_{ij} = 0$  otherwise. It is symmetric (i.e.,  $A_{ij} = A_{ji}$ ) for undirected networks. Function  $\delta$  is the Kronecker delta, and  $c_i$  represents the index of the community to which node  $i$  belongs. In equation (1), the frequency of links within communities, corresponding to  $A_{ij} \delta_{c_i, c_j}$  in equation (1), is compared against that for the null model, corresponding to  $k_i k_j \delta_{c_i, c_j} / 2M$ . Under the null model, the expected number of links between nodes  $i$  and  $j$  is equal to  $P_{ij}^{\text{und}} = k_i k_j / 2M$ . In other words, the null model is the configuration model in which each node has the same degree as in the original network.

#### 3.2 Modularity for directed networks

The modularity for directed networks is defined by:

$$Q^{\text{dir}} \equiv \frac{1}{M} \sum_{i,j=1}^N \left( A_{ij} - \frac{k_i^{\text{in}} k_j^{\text{out}}}{M} \right) \delta_{c_i, c_j}, \quad (2)$$

where  $k_i^{\text{in}}$  and  $k_j^{\text{out}}$  are the in-degree of node  $i$  and the out-degree of node  $j$ , respectively [36,45]. In general,  $A$  is allowed to be asymmetric, and  $A_{ij} = 1$  if there is a link from node  $j$  to  $i$  and  $A_{ij} = 0$  otherwise. In  $Q^{\text{dir}}$ , the number of intra-community links is compared against the null model in which the expected number of links emanating from node  $j$  to  $i$  is equal to  $P_{ij}^{\text{dir}} = k_i^{\text{in}} k_j^{\text{out}} / M$ . In other words, the null model is the directed variant of the configuration model in which each node has the same in- and out-degrees as in the original network.

#### 3.3 Modularity for DAGs

Although it is possible to apply  $Q^{\text{und}}$  and  $Q^{\text{dir}}$  to detect communities in DAGs, neither of them incorporates the partial order imposed on nodes in DAGs. Therefore, we propose a null model for DAGs by generalizing the random DAG model proposed in references [22,23] (for other null models of DAGs, see Ref. [46]). In the random DAG model, links are randomized while preserving the in- and out-degree of each node and the order of nodes as specified by  $\ell_i$ . The difference between the present definition and that in references [22,23] is that the former allows a layer to contain multiple nodes, whereas each node constitutes a single layer in the latter case.

Following the derivation in references [22,23], we calculate the expected number of links  $P_{ij}^{\text{dag}}$  from node  $j$  to node  $i$ . We start by defining

$$\mu_\ell \equiv \sum_{i: 1 \leq \ell_i < \ell} k_i^{\text{in}} - \sum_{i: 1 \leq \ell_i < \ell} k_i^{\text{out}}, \quad (3)$$

$$\lambda_\ell \equiv \sum_{i: 1 \leq \ell_i < \ell} k_i^{\text{in}} - \sum_{i: 1 \leq \ell_i \leq \ell} k_i^{\text{out}} = \mu_\ell - \sum_{i: \ell_i = \ell} k_i^{\text{out}}, \quad (4)$$

where  $1 \leq \ell \leq L$ . In words,  $\mu_\ell$  is equal to the number of links from nodes in layers  $\{\ell, \ell + 1, \dots, L\}$  to nodes in layers  $\{1, 2, \dots, \ell - 1\}$ , and  $\lambda_\ell$  is equal to the number of links from nodes in layers  $\{\ell + 1, \ell + 2, \dots, L\}$  to nodes in layers  $\{1, 2, \dots, \ell - 1\}$ . Graphically,  $\lambda_\ell$  and  $\mu_\ell$  are equal to the number of links crossing a section at  $\ell$  (dotted line in Fig. 1) and one before layer  $\ell$  (dashed line in Fig. 1), respectively. In the example shown in Figure 1,  $\lambda_\ell = 1$  and  $\mu_\ell = 4$ .

The in-degree of node  $i$ ,  $k_i^{\text{in}}$ , can be visualized as  $k_i^{\text{in}}$  in-stubs attached to node  $i$ , where a stub is a half-link. Likewise for the out-degree. Joining an in-stub and an out-stub yields a link. We calculate the expected number for an in-stub attached to node  $i$  to connect to node  $j$ , which we denote by  $p_{ij}$ . Let us first consider the case  $\ell_j = \ell_i + 1$ . In this case, we obtain

$$p_{ij} = \frac{k_j^{\text{out}}}{\mu_{\ell_i + 1}} \quad (5)$$

because  $j$  possesses  $k_j^{\text{out}}$  out-stubs and there are  $\mu_{\ell_i + 1}$  out-stubs from which an in-stub attached to node  $i$  can choose. The probability that an in-stub of node  $i$  does not

connect to a node in layer  $\ell_i + 1$  is given by:

$$1 - \sum_{j: \ell_j = \ell_i + 1} p_{ij} = 1 - \frac{\sum_{j: \ell_j = \ell_i + 1} k_j^{\text{out}}}{\mu_{\ell_i + 1}} = \frac{\lambda_{\ell_i + 1}}{\mu_{\ell_i + 1}}. \quad (6)$$

Now we assume in general that node  $j$  belongs to any layer  $\ell_j > \ell_i$ . Then, an in-stub of node  $i$  can connect to node  $j$  only if it does not connect to a node in layers  $\ell_i < \ell < \ell_j$ . By iteratively applying the argument leading to equation (6), we find that the probability that an in-stub of node  $i$  does not connect to any node in layers  $\ell_i < \ell < \ell_j$  is given by:

$$\prod_{\ell = \ell_i + 1}^{\ell_j - 1} \frac{\lambda_{\ell_j}}{\mu_{\ell_j}}. \quad (7)$$

Provided that the in-stub of node  $i$  does not connect to a node in layers  $\ell_i < \ell < \ell_j$ , the probability that it connects to node  $j$  is given by:

$$\frac{k_j^{\text{out}}}{\mu_{\ell_j}}. \quad (8)$$

Using equations (7) and (8), we obtain

$$p_{ij} = k_j^{\text{out}} \frac{\prod_{\ell = \ell_i + 1}^{\ell_j - 1} \lambda_{\ell_j}}{\prod_{\ell = \ell_i + 1}^{\ell_j} \mu_{\ell_j}}. \quad (9)$$

Because node  $i$  has  $k_i^{\text{in}}$  in-stubs, the expected number of links from  $j$  to  $i$  is given by:

$$P_{ij}^{\text{dag}} = k_i^{\text{in}} k_j^{\text{out}} \frac{\prod_{\ell = \ell_i + 1}^{\ell_j - 1} \lambda_{\ell_j}}{\prod_{\ell = \ell_i + 1}^{\ell_j} \mu_{\ell_j}}. \quad (10)$$

Using  $P_{ij}^{\text{dag}}$ , we define the modularity for DAGs, denoted by  $Q^{\text{dag}}$ , by:

$$Q^{\text{dag}} \equiv \frac{1}{M} \sum_{i,j} \left( A_{ij} - P_{ij}^{\text{dag}} \right) \delta_{c_i, c_j}, \quad (11)$$

where  $P_{ij}^{\text{dag}}$  is given by:

$$P_{ij}^{\text{dag}} \equiv \begin{cases} 0 & (\ell_i \geq \ell_j), \\ k_i^{\text{in}} k_j^{\text{out}} \frac{\prod_{\ell = \ell_i + 1}^{\ell_j - 1} \lambda_{\ell}}{\prod_{\ell = \ell_i + 1}^{\ell_j} \mu_{\ell}} & (\ell_i < \ell_j). \end{cases} \quad (12)$$

## 4 Spectral methods for modularity maximization

Because maximization of modularity is a combinationally hard problem, many approximate algorithms have been proposed for maximizing  $Q^{\text{und}}$  [2], whose variants are also used to maximize  $Q^{\text{dir}}$  [2,36,47]. Here we adapt the spectral method [48,49] to the case of  $Q^{\text{dag}}$ .

### 4.1 Spectral method for undirected networks

Before describing the spectral method for  $Q^{\text{dag}}$ , we review the method for  $Q^{\text{und}}$  [48,49]. The spectral method realizes community detection by iteratively bipartitioning a tentative community.

We initially partition the entire network into two groups of nodes. To this end, we use the fact that  $Q^{\text{und}}$  is written in matrix form as:

$$Q^{\text{und}} = \frac{1}{4M} \mathbf{s}^\top B^{\text{und}} \mathbf{s}, \quad (13)$$

where  $\top$  denotes the transposition,  $\mathbf{s} = (s_1 \cdots s_N)^\top$ ,  $s_i \in \{-1, 1\}$ , and  $B^{\text{und}}$  is an  $N \times N$  symmetric matrix whose elements are given by  $B_{ij}^{\text{und}} \equiv A_{ij} - (k_i k_j / 2M)$ . Node  $i$  is classified to the first and second groups if  $s_i = -1$  and  $s_i = 1$ , respectively.

Finding the vector  $\mathbf{s}$  that maximizes  $Q^{\text{und}}$  is an NP-complete problem [50]. A commonly applied heuristics is to relax elements of  $\mathbf{s}$  to take continuous values and impose the normalization constraint  $\mathbf{s}^\top \mathbf{s} = N$ . Then,  $Q^{\text{und}}$  is maximized when  $\mathbf{s}$  is the eigenvector associated with the largest eigenvalue of  $B^{\text{und}}$ , which we denote by  $\mathbf{u}$ . We carry out bipartition by setting  $s_i = \text{sgn}(u_i)$  ( $1 \leq i \leq N$ ).

The spectral method takes advantage of the fact that we can rapidly calculate  $\mathbf{u}$  using the power method. In fact, the power method requires calculation of the product  $B^{\text{und}} \mathbf{x}$  for changing  $\mathbf{x}$ . Although this computation may look computationally costly because  $B^{\text{und}}$  is a dense matrix,  $B^{\text{und}} \mathbf{x}$  can be expressed as:

$$B^{\text{und}} \mathbf{x} = A \mathbf{x} - \frac{\mathbf{k}(\mathbf{k}^\top \mathbf{x})}{2M}, \quad (14)$$

where  $\mathbf{k} \equiv (k_1, k_2, \dots, k_N)^\top$  [48,49]. The inner product  $\mathbf{k}^\top \mathbf{x}$  is calculated in time  $O(N)$ , while the calculation of  $A \mathbf{x}$  requires time  $O(N + M)$ . For sparse graphs,  $M = O(N)$  such that  $B^{\text{und}} \mathbf{x}$  can be calculated in  $O(N)$  time, accelerating the power method.

Once the network is partitioned into two tentative communities, we repeat selecting and bipartitioning one tentative community until  $Q^{\text{und}}$  stops increasing. To decide whether to approve a proposed bipartition of a tentative community  $C$  into two groups of nodes  $C_1$  and  $C_2$ , one needs to calculate the change in modularity by the partitioning, which is given by:

$$\Delta Q^{\text{und}} = \frac{1}{2M} \left[ \sum_{i \in C_1, j \in C_2} (B_{ij}^{\text{und}} + B_{ji}^{\text{und}}) - \sum_{i,j \in C} B_{ij}^{\text{und}} \right]. \quad (15)$$

In matrix form,  $\Delta Q^{\text{und}}$  is written as:

$$\Delta Q^{\text{und}} = \frac{1}{4M} \tilde{\mathbf{s}}^\top \tilde{B}^{\text{und}} \tilde{\mathbf{s}}, \quad (16)$$

where  $\tilde{\mathbf{s}}$  is a column vector with length  $|C|$ , and  $\tilde{B}^{\text{und}}$  is a  $|C| \times |C|$  symmetric matrix whose elements are defined



by  $\tilde{B}_{ij}^{\text{und}} \equiv B_{ij}^{\text{und}} - \delta_{ij} \sum_{k \in C} B_{ik}^{\text{und}}$  [48]. When  $C$  is the entire network,  $\Delta Q^{\text{und}}$ ,  $\tilde{B}^{\text{und}}$ , and  $\tilde{\mathbf{s}}$  are equal to  $Q^{\text{und}}$ ,  $B^{\text{und}}$ , and  $\mathbf{s}$ , respectively. Under the constraint  $\tilde{\mathbf{s}}^\top \tilde{\mathbf{s}} = |C|$ , equation (16) is maximized if  $\tilde{\mathbf{s}}$  is the eigenvector associated with the largest eigenvalue of  $\tilde{B}^{\text{und}}$ , which we denote by  $\tilde{\mathbf{u}}$ . Finally, we bipartition  $C$  by setting  $\tilde{s}_i = \text{sgn}(\tilde{u}_i)$  ( $i \in C$ ).

## 4.2 Spectral method for directed networks

The spectral method for directed networks was implemented in reference [36]. Here we briefly recapitulate their arguments. We can rewrite the modularity for directed network as  $Q^{\text{dir}} = (1/4M) \mathbf{s}^\top [B^{\text{dir}} + (B^{\text{dir}})^\top] \mathbf{s}$  with  $(B^{\text{dir}})_{ij} \equiv A_{ij} - (k_i^{\text{in}} k_j^{\text{out}})/M$ . Because  $B^{\text{dir}} + (B^{\text{dir}})^\top$  is a symmetric matrix, we can maximize  $Q^{\text{dir}}$  by replacing  $B^{\text{und}}$  in equation (14) by  $B^{\text{dir}} + (B^{\text{dir}})^\top$  and applying the same spectral method as that for  $Q^{\text{und}}$ .

To calculate the change in modularity  $\Delta Q^{\text{dir}}$  caused by bipartitioning, we replace  $B^{\text{und}}$  by  $B^{\text{dir}} + (B^{\text{dir}})^\top$  in equations (15) and (16). Then, we maximize  $\Delta Q^{\text{dir}}$  by following the same steps as for maximizing  $\Delta Q^{\text{und}}$ .

## 4.3 Spectral method for DAGs

The application of the spectral method to  $Q^{\text{dag}}$  is straightforward. For  $Q^{\text{dag}}$ , a single step in the power method can be accelerated as in the case of  $Q^{\text{und}}$  and  $Q^{\text{dir}}$  as follows.

First, we write

$$Q^{\text{dag}} = \frac{1}{4M} \mathbf{s}^\top [B^{\text{dag}} + (B^{\text{dag}})^\top] \mathbf{s}, \quad (17)$$

where  $B_{ij}^{\text{dag}} \equiv A_{ij} - P_{ij}^{\text{dag}}$ . To find a decomposition of  $[B^{\text{dag}} + (B^{\text{dag}})^\top] \mathbf{x}$  similar to equation (14), we write

$$P_{ij}^{\text{dag}} = k_i^{\text{in}} k_j^{\text{out}} f(\ell_i, \ell_j), \quad (18)$$

where

$$f(\ell_i, \ell_j) \equiv \begin{cases} 0 & (\ell_i \geq \ell_j), \\ \frac{\prod_{\ell=\ell_i+1}^{\ell_j-1} \lambda_\ell}{\prod_{\ell=\ell_i+1}^{\ell_j} \mu_\ell} & (\ell_i < \ell_j). \end{cases} \quad (19)$$

As discussed in detail in reference [23], equation (19) when  $\ell_i < \ell_j$  is identical to:

$$f(\ell_i, \ell_j) = \begin{cases} \frac{f(\ell_i, \ell^{\text{up}}(j)) f(\ell^{\text{low}}(i), \ell_j)}{f(\ell^{\text{low}}(i), \ell^{\text{up}}(j))} & (\lambda_{\ell_i+1}, \dots, \lambda_{\ell_j-1} \neq 0), \\ 0 & (\text{otherwise}), \end{cases} \quad (20)$$

where,  $\ell^{\text{low}}(i)$  is the largest value of  $\ell (\leq \ell_i)$  such that  $\lambda_\ell = 0$ , and  $\ell^{\text{up}}(j)$  is the smallest value of  $\ell (\geq \ell_j)$  such that  $\lambda_\ell = 0$ . If there is no  $\ell (\leq \ell_i)$  or  $\ell (\geq \ell_j)$  such that  $\lambda_\ell = 0$ , we obtain  $\ell^{\text{low}}(i) = 1$  and  $\ell^{\text{up}}(j) = L$ , respectively. In words,  $f(\ell_i, \ell_j) > 0$  if and only if for every layer between  $\ell_i$  and  $\ell_j$ , we find at least one link penetrating the layer.

Using equation (20), we decompose  $P_{ij}^{\text{dag}}$  in the case  $\ell_i < \ell_j$  and  $\lambda_{\ell_i+1}, \dots, \lambda_{\ell_j-1} \neq 0$  as:

$$P_{ij}^{\text{dag}} = \frac{f(\ell_i, \ell^{\text{up}}(j))}{f(\ell^{\text{low}}(i), \ell^{\text{up}}(j))} k_i^{\text{in}} \cdot f(\ell^{\text{low}}(i), \ell_j) k_j^{\text{out}}. \quad (21)$$

It should be noted that  $P_{ij}^{\text{dag}} = 0$  otherwise. Because  $\lambda_{\ell_i+1}, \dots, \lambda_{\ell_j-1} \neq 0$  if and only if  $\ell^{\text{low}}(i) = \ell^{\text{low}}(j)$  and  $\ell^{\text{up}}(i) = \ell^{\text{up}}(j)$ , we can rewrite equation (21) as:

$$P_{ij}^{\text{dag}} = \frac{f(\ell_i, \ell^{\text{up}}(i))}{f(\ell^{\text{low}}(i), \ell^{\text{up}}(i))} k_i^{\text{in}} \cdot f(\ell^{\text{low}}(j), \ell_j) k_j^{\text{out}}. \quad (22)$$

For each layer  $\ell$ , we define  $\boldsymbol{\kappa}^{\text{in}}(\ell) = (\kappa_1^{\text{in}}(\ell) \dots \kappa_N^{\text{in}}(\ell))^\top$  and  $\boldsymbol{\kappa}^{\text{out}}(\ell) = (\kappa_1^{\text{out}}(\ell) \dots \kappa_N^{\text{out}}(\ell))^\top$  by:

$$\kappa_i^{\text{in}}(\ell) = \begin{cases} 0 & (\ell_i \neq \ell), \\ \frac{f(\ell_i, \ell^{\text{up}}(i))}{f(\ell^{\text{low}}(i), \ell^{\text{up}}(i))} k_i^{\text{in}} & (\ell_i = \ell), \end{cases} \quad (23)$$

and

$$\kappa_i^{\text{out}}(\ell) = \begin{cases} 0 & (\ell < \ell^{\text{low}}(i) \text{ or } \ell_i \leq \ell), \\ f(\ell^{\text{low}}(i), \ell_i) k_i^{\text{out}} & (\ell^{\text{low}}(i) \leq \ell < \ell_i), \end{cases} \quad (24)$$

respectively. By combining equations (22), (23) and (24), we obtain

$$P_{ij}^{\text{dag}} = \kappa_i^{\text{in}}(\ell_i) \kappa_j^{\text{out}}(\ell_i) = \sum_{\ell=1}^L \kappa_i^{\text{in}}(\ell) \kappa_j^{\text{out}}(\ell). \quad (25)$$

Finally, we decompose  $B^{\text{dag}} \mathbf{x}$  as:

$$B^{\text{dag}} \mathbf{x} = A \mathbf{x} - \sum_{\ell=1}^L \boldsymbol{\kappa}^{\text{in}}(\ell) (\boldsymbol{\kappa}^{\text{out}}(\ell)^\top \mathbf{x}). \quad (26)$$

Therefore, once  $\{\boldsymbol{\kappa}^{\text{in}}(\ell), \boldsymbol{\kappa}^{\text{out}}(\ell)\}_{1 \leq \ell \leq L}$  have been calculated beforehand, the computation of  $B^{\text{dag}} \mathbf{x}$  takes  $O(LN)$  time, and likewise for  $(B^{\text{dag}})^\top \mathbf{x}$ .

To implement repeated bipartitioning, we maximize the change in modularity,  $\Delta Q^{\text{dag}}$ , which a bipartition of a community causes. We calculate  $\Delta Q^{\text{dag}}$  by replacing  $B^{\text{und}}$  by  $B^{\text{dag}} + (B^{\text{dag}})^\top$  in equations (15) and (16). Then, we follow the same steps as for maximizing  $\Delta Q^{\text{und}}$ .

## 5 Results

In this section, we test the proposed spectral method for maximizing  $Q^{\text{dag}}$  for several DAG data. We test the method against the spectral method maximizing  $Q^{\text{und}}$  and  $Q^{\text{dir}}$  and the so-called Louvain method, which is a non-spectral algorithm that heuristically maximizes  $Q^{\text{und}}$  [51]. We use the implementation of the Louvain method in the igraph package of R [52]. In our implementation of the spectral method, we add a fine tuning step after every bipartitioning in a similar fashion to references [48,49] (Appendix A). We do not apply this fine tuning to the Louvain

method because this fine tuning is specialized to the spectral method. We also apply a postprocessing procedure after the spectral method or the Louvain method has terminated (Appendix B). To apply the spectral method and the Louvain method to maximize  $Q^{\text{und}}$ , and also to calculate  $Q^{\text{und}}$  for the final partition obtained by various methods, we ignore link direction, or equivalently, use  $A + A^T$  as the adjacency matrix of the undirected network.

To quantify similarity between two partitions of the same network, we calculate the Jaccard index for each pair of partitions [2]. The Jaccard index is defined by:

$$J \equiv \frac{a_1}{a_1 + a_0}, \quad (27)$$

where  $a_0$  and  $a_1$  are the number of node pairs that are classified to the same community in only one partition and in both partitions, respectively. If  $J = 1$ , the two partitions are exactly the same. If  $J = 0$ , they completely disagree.

### 5.1 Citation networks

As an example of temporal networks represented by DAGs, we study two citation networks of articles posted on the e-print archive [arXiv.org](http://arxiv.org). We use data on the High Energy Physics Phenomenology (HepPh) and High Energy Physics Theory (HepTh) sections of arXiv, which exhaustively cover the citations between January 1993 and April 2003 [53–55].

The HepPh and HepTh data contain 34 546 and 27 770 nodes, and 421 578 and 352 807 links, respectively. There are four mutually exclusive types of nodes and links excluded from the following analysis. First, we discard 44 and 39 self-loops in the HepPh and HepTh networks, respectively. Second, we discard 2622 and 1475 links in HepPh and HepTh networks, respectively, which contradict the arrow of time (i.e., articles citing others newer than themselves). Third, in the publicly available data of the HepPh network, the information about the publication date is not provided for the articles posted after 11 March 2002. Because we cannot assign a date-based layer  $\ell_i$  to these articles, we remove 3985 such articles. Fourth, we remove 74 246 links in the HepPh network that are incident to at least one node whose date information is missing. Although we have removed a fraction  $74\,246/421\,578 = 0.176$  of the links according to the fourth criterion, all removed links are those incident to the newest nodes. Therefore, the network restricted to the dates before 11 March 2002, which we effectively consider in the following, is not affected by the removal of these links.

After removing self-loops, links contradicting the arrow of time, nodes without the information about the publication date, and the links incident to these nodes, we obtain 30 561 nodes and 344 666 links in the HepPh network and 27 770 nodes and 351 293 links in the HepTh network. We use the largest weakly connected component of each network. Layers are defined by the date of publication. It should be noted that some layers had no node

**Table 1.** Properties of the networks analyzed in Section 5. The number of nodes, that of links, and that of layers are denoted by  $N$ ,  $M$ , and  $L$ , respectively. These values refer to those of the largest weakly connected component of each network.

Data	$N$	$M$	$L$
HepPh	30 337	344 578	3683
HepTh	27 377	351 033	4139
<i>E. coli</i>	328	456	5
C1	20	29	4
C2	32	55	7
C3	48	134	8
C4	70	158	7
C6	64	137	8

**Table 2.** Modularity values for the citation networks. The results obtained by maximization of  $Q^{\text{und}}$ ,  $Q^{\text{dir}}$ , and  $Q^{\text{dag}}$  using the spectral method are shown in the rows labelled “s-und”, “s-dir”, and “s-dag”, respectively. The results obtained by maximization of  $Q^{\text{und}}$  using the Louvain method are shown in the rows labeled “L-und”.  $N_c$  denotes the number of communities.

Data	Method	$N_c$	$Q^{\text{und}}$	$Q^{\text{dir}}$	$Q^{\text{dag}}$
HepPh	s-und	19	0.7292	0.7292	0.7259
	s-dir	18	0.7288	0.7288	0.7257
	s-dag	16	0.7300	0.7300	0.7263
	L-und	19	<b>0.7331</b>	<b>0.7332</b>	<b>0.7292</b>
HepTh	s-und	21	<b>0.6569</b>	<b>0.6571</b>	<b>0.6332</b>
	s-dir	30	0.6447	0.6450	0.6207
	s-dag	20	0.6452	0.6453	0.6320
	L-und	28	0.6558	0.6561	0.6266

because no article was published on that date. This fact does not change the following results because  $Q^{\text{dag}}$  and its optimization procedure described in the previous sections are not affected by empty layers. Some basic quantities of the largest weakly connected component of the two networks are summarized in Table 1.

Modularity values obtained after the maximization procedure are summarized for the two networks in Table 2. A row corresponds to a combination of the maximized modularity measure (i.e.,  $Q^{\text{und}}$ ,  $Q^{\text{dir}}$ , or  $Q^{\text{dag}}$ ) and the maximization method (i.e., spectral or Louvain). For example, s-und represents the spectral method for maximizing  $Q^{\text{und}}$ . The columns correspond to the three modularity values measured at the end of the maximization procedure, given a modularity maximization method corresponding to a row.

The modularity values shown in Table 2 seem to be large. It turns out that, in both networks, a partitioning method designed for  $Q^{\text{und}}$  (i.e., L-und for the HepPh network and s-und for the HepTh network) has yielded the largest value for not only  $Q^{\text{und}}$  but also  $Q^{\text{dir}}$  and  $Q^{\text{dag}}$ . Therefore, for these networks, it is better to use a method designed for maximizing  $Q^{\text{und}}$  to (also) maximize  $Q^{\text{dag}}$  than a method designed for maximizing  $Q^{\text{dag}}$ . Table 2 also indicates that the spectral method designed for maximizing  $Q^{\text{dag}}$  (s-dag) does not fall far behind s-und or L-und

**Table 3.** Jaccard indices between the partitions obtained for the citation networks.

Data	Method	s-und	s-dir	s-dag	L-und
HepPh	s-und	1	0.6383	0.4116	0.4287
	s-dir	0.6383	1	0.4298	0.3905
	s-dag	0.4116	0.4298	1	0.4557
	L-und	0.4287	0.3905	0.4557	1
HepTh	s-und	1	0.3154	0.2918	0.2859
	s-dir	0.3154	1	0.3417	0.3262
	s-dag	0.2918	0.3417	1	0.2804
	L-und	0.2859	0.3262	0.2804	1

in terms of the obtained  $Q^{\text{dag}}$  value. In fact, regardless of the method, we obtained similar values of  $Q^{\text{und}}$ ,  $Q^{\text{dir}}$ , and  $Q^{\text{dag}}$ .

The similarity between each pair of the four optimized partitions for each network (Tab. 2) is shown in Table 3, where similarity is measured in terms of the Jaccard index. We find that the Jaccard indices are not very large, in particular for the HepTh network, although the different partitioning methods have yielded close modularity values, as shown in Table 2. Nevertheless, the Jaccard index may not be very large even between partitions with close modularity values obtained from the same stochastic partitioning algorithm applied to the same network. For example, the Jaccard index values between 0.5 and 0.9 reported in reference [56] are larger than but comparable with the present values. It has also been reported that partitioning results obtained from the same network yielding close modularity values can be fairly different [2,57].

## 5.2 Transcriptional regulation network

We study the compartmentalization of the transcriptional regulation network of the bacteria *Escherichia coli* (*E. coli*) using publicly available data [58,59]. Nodes of this network are operons. Links are directed from the operon that encodes a transcription factor to an operon regulated by the transcription factor. The network, which is a DAG, contains 423 nodes and 519 directed links without self loops. We extracted the largest weakly connected component, which consisted of 328 nodes and 456 links. Layers were determined by a leaf-removal algorithm [43]. In this algorithm, the nodes with out-degree zero are classified to the lowest layer. Then, these nodes together with links connecting to these nodes are removed. The nodes with out-degree zero in the remaining network are classified to next lowest layer. This procedure is iterated until all nodes are exhausted. The basic quantities of this network are summarized in Table 1.

The modularity values obtained from the four modularity maximization algorithms are shown in Table 4. Algorithm s-und produces the largest  $Q^{\text{und}}$ ,  $Q^{\text{dir}}$ , and  $Q^{\text{dag}}$ , similarly to the results for the HepTh network (Tab. 2). The Jaccard indices are shown in Table 5. They are much larger than those for the citation networks (Tab. 3), suggesting that the partitions obtained by the different

**Table 4.** Modularity values for the *E. coli* transcriptional regulation network. See the caption of Table 2 for legends.

Data	Method	$N_c$	$Q^{\text{und}}$	$Q^{\text{dir}}$	$Q^{\text{dag}}$
<i>E. coli</i>	s-und	13	<b>0.7366</b>	<b>0.7373</b>	<b>0.6998</b>
	s-dir	13	0.7325	0.7332	0.6938
	s-dag	13	0.7136	0.7142	0.6988
	L-und	13	0.7357	0.7364	0.6972

**Table 5.** Jaccard indices between the partitions obtained for the *E. coli* transcriptional regulation network.

Data	Method	s-und	s-dir	s-dag	L-und
<i>E. coli</i>	s-und	1	0.7885	0.6747	0.8875
	s-dir	0.7885	1	0.5998	0.7927
	s-dag	0.6747	0.5998	1	0.6699
	L-und	0.8875	0.7927	0.6699	1

methods are relatively similar to each other for the present network.

## 5.3 Dominance networks

As a final example, we examine DAGs induced by dominance hierarchies in ant colonies using the data in reference [18]. The data set contains aggression-based hierarchy among workers in six ant colonies of species *Diacamma sp.* Nodes represent individual female ant workers and a link is drawn from the attacking to the attacked ants, where an attack is defined as a bite and jerk. Layers are determined by the leaf-removal algorithm used for the transcriptional regulation network. The data set contains six colonies, five of whose largest weakly connected components are DAGs that we use in the following analysis. Basic quantities of the largest weakly connected component of the five DAGs are summarized in Table 1.

The present DAGs are smaller than those analyzed in the previous sections. Therefore, for some networks, we were able to calculate the partition that exactly maximized the modularity using integer linear programming [50]. In this method, originally applied to  $Q^{\text{und}}$ , variables  $X_{ij} \in \{0,1\}$  are defined for every pair of nodes  $i$  and  $j$ , where  $X_{ij} = 1$  ( $X_{ij} = 0$ ) indicates that  $i$  and  $j$  are classified to the same community (different communities). Then, we rewrite the modularity in terms of  $X_{ij}$  by replacing  $\delta_{c_i c_j}$  by  $X_{ij}$  in equations (1), (2), and (11) for  $Q^{\text{und}}$ ,  $Q^{\text{dir}}$ , and  $Q^{\text{dag}}$ , respectively. The variables need to satisfy  $X_{ii} = 1$ ,  $X_{ij} = X_{ji}$ , and  $X_{ij} + X_{jh} - 2X_{ih} \leq 1$  for all  $1 \leq h, i, j \leq N$ , which defines an integer linear program for exact maximization of modularity.

The results for modularity maximization and the Jaccard indices for the obtained partitions are shown in Tables 6 and 7, respectively. Table 6 indicates that the obtained modularity values vary across colonies (i.e., networks) and do not solely depend on the size of the network. Therefore, different colonies may have different degrees of community structure. The table also indicates that s-dag

**Table 6.** Modularity values for five dominance networks. The results obtained with the optimal modularity maximization method are shown in rows labeled “o-und”, “o-dir”, and “o-dag”. In the table, “—” indicates that the optimal method has not terminated. See the caption of Table 2 for the other legends.

Data	Method	$N_c$	$Q^{\text{und}}$	$Q^{\text{dir}}$	$Q^{\text{dag}}$
C1	s-und	4	0.2990	0.3258	0.3393
	s-dir	3	0.2943	0.3317	0.3421
	s-dag	3	0.2812	0.3282	0.3468
	L-und	4	<b>0.3288</b>	0.3401	0.3633
	o-und	4	<b>0.3288</b>	0.3401	0.3633
	o-dir	4	0.3157	<b>0.3484</b>	<b>0.3662</b>
	o-dag	4	0.3157	<b>0.3484</b>	<b>0.3662</b>
C2	s-und	4	0.3522	0.3593	0.3729
	s-dir	4	<b>0.3607</b>	0.3702	<b>0.3951</b>
	s-dag	4	<b>0.3607</b>	0.3702	<b>0.3951</b>
	L-und	4	<b>0.3607</b>	0.3702	<b>0.3951</b>
	o-und	4	<b>0.3607</b>	0.3702	<b>0.3951</b>
	o-dir	5	0.3526	<b>0.3716</b>	0.3890
	o-dag	4	<b>0.3607</b>	0.3702	<b>0.3951</b>
C3	s-und	5	0.2339	0.2372	0.2503
	s-dir	5	0.2311	<b>0.2594</b>	0.2643
	s-dag	6	<b>0.2416</b>	0.2531	<b>0.2654</b>
	L-und	3	0.2378	0.2421	0.2572
	o-und	—	—	—	—
	o-dir	5	0.2311	<b>0.2594</b>	0.2643
	o-dag	6	<b>0.2416</b>	0.2531	<b>0.2654</b>
C4	s-und	5	<b>0.2690</b>	0.2782	0.2882
	s-dir	6	0.2612	<b>0.2788</b>	0.2899
	s-dag	5	0.2604	0.2767	<b>0.2914</b>
	L-und	5	0.2626	0.2656	0.2766
	o-und	—	—	—	—
	o-dir	—	—	—	—
	o-dag	—	—	—	—
C6	s-und	6	0.3374	0.3396	0.3512
	s-dir	5	0.3308	0.3422	0.3412
	s-dag	6	0.3348	0.3428	0.3600
	L-und	5	0.3467	0.3504	0.3493
	o-und	5	<b>0.3487</b>	<b>0.3529</b>	0.3521
	o-dir	5	<b>0.3487</b>	<b>0.3529</b>	0.3521
	o-dag	6	0.3356	0.3440	<b>0.3606</b>

realizes the largest  $Q^{\text{dag}}$  values in all colonies except C1, among the four heuristic methods (i.e., s-und, s-dir, s-dag, and L-und). Algorithm s-dag realizes the optimal  $Q^{\text{dag}}$  value obtained by o-dag for C2 and C3. The  $Q^{\text{dag}}$  value obtained by s-dag is also close to that obtained by o-dag for C6, whereas this is not the case for C1. Similarity among the partitions obtained by the different algorithms depends on colonies without clear patterns (Tab. 7).

## 6 Conclusions

We have proposed a modularity measure for DAGs and a spectral algorithm to maximize it by extending the spectral algorithm developed for undirected networks. We found that the obtained modularity values are rather

independent of whether we used the proposed algorithm or other algorithms known for less restricted networks such as undirected or directed networks allowing for cycles. Therefore, up to our numerical efforts, simply applying modularity maximization methods for undirected networks to DAGs may be practically innocuous. We stress that we have reached this conclusion by actually developing a modularity measure for DAGs and testing it against previous methods using several data sets.

The spectral method was presented as an example heuristic for maximizing the proposed modularity measure. The proposed modularity measure can be also maximized by other approximate methods, which may surpass the performance of the spectral method.

We acknowledge Steve Gregory and Kohei Tamura for discussion and careful reading of the manuscript. L.S. acknowledges the support provided through DAAD. N.M. acknowledges the support provided through JST, CREST, and JST, ERATO, Kawarabayashi Large Graph Project. T.T. and N.M. designed the research. L.S., T.T., and N.M. developed the theory. L.S. and T.T. analyzed the data. L.S., T.T., and N.M. discussed the results and wrote the paper.

L. Speidel and T. Takaguchi contributed equally to the work.

## Appendix A: Fine tuning after bipartitioning

In the spectral methods for maximizing  $Q^{\text{und}}$ ,  $Q^{\text{dir}}$ , and  $Q^{\text{dag}}$ , we carried out the following fine tuning procedure every time after a community was bipartitioned. Suppose that we have bipartitioned a community  $C$  into two communities  $C_1$  and  $C_2$ . For every node in  $C$ , we tentatively moved the node to the opposite community and calculated the change in the targeted modularity value. Then, we adopted the attempted move that maximized the modularity under the condition that the modularity increased by the move. We repeated this procedure under the constraint that each node was moved at most once, until no further increase in modularity was possible. This fine tuning procedure slightly modifies that proposed for  $Q^{\text{und}}$  [48,49]. We modified the original procedure because it involved repetition of a procedure similar to the aforementioned one and did not terminate on our data when we attempted to maximize  $Q^{\text{dag}}$ .

## Appendix B: Postprocessing

After the completion of modularity maximization with the spectral or Louvain method, we carried out the following postprocessing algorithm to enhance the targeted modularity value. Our postprocessing algorithm resembles that employed in other modularity maximization algorithms [51,60].

Step 1: For a given node  $i$  ( $1 \leq i \leq N$ ), we tentatively moved it to the community to which each adjacent node of  $i$  belonged. The move attempt that increased the modularity by the largest amount was adopted. We scanned the  $N$  nodes in random order. We neglected the direction of link



**Table 7.** Jaccard indices between the partitions obtained for the dominance networks.

Data	Method	s-und	s-dir	s-dag	L-und	o-und	o-dir	o-dag
C1	s-und	1	0.7288	0.6452	0.5965	0.5965	0.5439	0.5439
	s-dir	0.7288	1	0.5946	0.4658	0.4658	0.4857	0.4857
	s-dag	0.6452	0.5946	1	0.5970	0.5970	0.7049	0.7049
	L-und	0.5965	0.4658	0.5970	1	1	0.8235	0.8235
	o-und	0.5965	0.4658	0.5970	1	1	0.8235	0.8235
	o-dir	0.5439	0.4857	0.7049	0.8235	0.8235	1	1
	o-dag	0.5439	0.4857	0.7049	0.8235	0.8235	1	1
C2	s-und	1	0.7083	0.7083	0.7083	0.7083	0.5753	0.7083
	s-dir	0.7083	1	1	1	1	0.7724	1
	s-dag	0.7083	1	1	1	1	0.7724	1
	L-und	0.7083	1	1	1	1	0.7724	1
	o-und	0.7083	1	1	1	1	0.7724	1
	o-dir	0.5753	0.7724	0.7724	0.7724	0.7724	1	0.7724
	o-dag	0.7083	1	1	1	1	0.7724	1
C3	s-und	1	0.2727	0.2755	0.3028	–	0.2727	0.2755
	s-dir	0.2727	1	0.6687	0.3275	–	1	0.6687
	s-dag	0.2755	0.6687	1	0.3548	–	0.6687	1
	L-und	0.3028	0.3275	0.3548	1	–	0.3275	0.3548
	o-und	–	–	–	–	1	–	–
	o-dir	0.2727	1	0.6687	0.3275	–	1	0.6687
	o-dag	0.2755	0.6687	1	0.3548	–	0.6687	1
C4	s-und	1	0.5726	0.5928	0.4807	–	–	–
	s-dir	0.5726	1	0.8047	0.6039	–	–	–
	s-dag	0.5928	0.8047	1	0.5827	–	–	–
	L-und	0.4807	0.6039	0.5827	1	–	–	–
	o-und	–	–	–	–	1	–	–
	o-dir	–	–	–	–	–	1	–
	o-dag	–	–	–	–	–	–	1
C6	s-und	1	0.5653	0.4153	0.5841	0.5697	0.5697	0.3254
	s-dir	0.5653	1	0.3055	0.5779	0.6641	0.6641	0.2852
	s-dag	0.4153	0.3055	1	0.3699	0.3121	0.3121	0.7118
	L-und	0.5841	0.5779	0.3699	1	0.7588	0.7588	0.2959
	o-und	0.5697	0.6641	0.3121	0.7588	1	1	0.3448
	o-dir	0.5697	0.6641	0.3121	0.7588	1	1	0.3448
	o-dag	0.3254	0.2852	0.7118	0.2959	0.3448	0.3448	1

when judging adjacency in the case of directed networks including DAGs.

Step 2: First, we selected the smallest community in terms of the number of nodes, called the seed community. Second, we tentatively merged the seed community with each of the remaining communities to which the seed community is directly connected by a link regardless of the direction. Third, we measured the modularity value. We adopted the merge attempt that yielded the largest modularity but only when the modularity increased by the merge. Then, among the community that had never been selected as seed community, we selected the smallest community as seed community and attempted merger. We repeated the same procedure until no further increase in the modularity occurred.

We applied step 1 and then step 2. We repeated the combination of the two steps ten times. We also verified that swapping the order of the two steps had little impacts on the final modularity value.

## References

1. P. Holme, J. Saramäki, Phys. Rep. **519**, 97 (2012)
2. S. Fortunato, Phys. Rep. **486**, 75 (2010)
3. D. Chakrabarti, R. Kumar, A. Tomkins, in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, 2006* (ACM, New York, 2006), p. 554
4. C. Tantipathananandh, T. Berger-Wolf, D. Kempe, in *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Jose, 2007* (ACM, New York, 2007), p. 717
5. V. Kawadia, S. Sreenivasan, Sci. Rep. **2**, 794 (2012)
6. P. Ronhovde, S. Chakrabarty, D. Hu, M. Sahu, K.K. Sahu, K.F. Kelton, N.A. Mauro, Z. Nussinov, Eur. Phys. J. E **34**, 105 (2011)
7. D.J. Fenn, M.A. Porter, M. McDonald, S. Williams, N.F. Johnson, N.S. Jones, Chaos **19**, 033119 (2009)
8. G. Palla, A.-L. Barabási, T. Vicsek, Nature **446**, 664 (2007)

9. L. Gauvin, A. Panisson, C. Cattuto, PLoS ONE **9**, e86028 (2014)
10. P.J. Mucha, T. Richardson, K. Macon, M.A. Porter, J.-P. Onnela, Science **328**, 876 (2010)
11. D.S. Bassett, M.A. Porter, N.F. Wymbs, S.T. Grafton, J.M. Carlson, P.J. Mucha, Chaos **23**, 013142 (2013)
12. M. Sarzynska, E.A. Leicht, G. Chowell, M.A. Porter, [arXiv:1407.6297](https://arxiv.org/abs/1407.6297) (2015)
13. S. Pandit, Y. Yang, V. Kawadia, S. Sreenivasan, N.V. Chawla, in *Proceedings of the 2011 IEEE First Network Science Workshop, West Point, New York, 2011* (IEEE, Los Alamitos, 2011), p. 173
14. J. Sun, C. Faloutsos, S. Papadimitriou, P.S. Yu, in *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Jose, 2007* (ACM, New York, 2007), p. 687
15. J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference* (Morgan Kaufmann Publishers Inc., San Francisco, 1988)
16. M.I. Jordan, Stat. Sci. **19**, 140 (2004)
17. S. Allesina, M. Pascual, Ecol. Lett. **12**, 652 (2009)
18. H. Shimoji, M.S. Abe, K. Tsuji, N. Masuda, J. R. Soc. Interface **11**, 20140599 (2014)
19. D.B. McDonald, D. Shizuka, Behav. Ecol. **24**, 511 (2012)
20. D.J. de Solla Price, Science **149**, 510 (1965)
21. E.A. Leicht, G. Clarkson, K. Shedden, M.E.J. Newman, Eur. Phys. J. B **59**, 75 (2007)
22. B. Karrer, M.E.J. Newman, Phys. Rev. Lett. **102**, 128701 (2009)
23. B. Karrer, M.E.J. Newman, Phys. Rev. E **80**, 046110 (2009)
24. J. Ott, *Analysis of Human Genetic Linkage* (JHU Press, Baltimore, 1999)
25. B.M.E. Moret, L. Nakhleh, T. Warnow, C.R. Linder, A. Tholse, A. Padolina, J. Sun, R. Timme, IEEE ACM T. Comput. Biol. Bioinform. **1**, 13 (2004)
26. F. Radicchi, S. Fortunato, A. Vespignani, in *Models of Science Dynamics, Understanding Complex Systems*, edited by A. Scharnhorst, K. Börner, P. Besselaar (Springer, Berlin, Heidelberg, 2012), Vol. 69, p. 233
27. R. Pfitzner, I. Scholtes, A. Garas, C.J. Tessone, F. Schweitzer, Phys. Rev. Lett. **110**, 198701 (2013)
28. D. Kempe, J. Kleinberg, A. Kumar, in *Proceedings of the thirty-second annual ACM Symposium on Theory of Computing, Portland, 2000* (ACM, New York, 2000), p. 504
29. V. Kostakos, Physica A **388**, 1007 (2009)
30. J. Hopcroft, O. Khan, B. Kulis, B. Selman, Proc. Natl. Acad. Sci. USA **101**, 5249 (2004)
31. P. Chen, S. Redner, J. Informetr. **4**, 278 (2010)
32. M. Rosvall, C.T. Bergstrom, PLoS ONE **5**, e8694 (2010)
33. C.P. Massen, J.P.K. Doye, Phys. Rev. E **71**, 046101 (2005)
34. M.E.J. Newman, Phys. Rev. E **70**, 056131 (2004)
35. Y. Kim, S.-W. Son, H. Jeong, Phys. Rev. E **81**, 016103 (2010)
36. E.A. Leicht, M.E.J. Newman, Phys. Rev. Lett. **100**, 118703 (2008)
37. M.J. Barber, Phys. Rev. E **76**, 066102 (2007)
38. R. Guimerà, M. Sales-Pardo, L.A.N. Amaral, Phys. Rev. E **76**, 036102 (2007).
39. T. Murata, in *Proceedings of the International Conference on Social Science and Engineering, Vancouver, 2009* (IEEE, Los Alamitos, 2009), Vol. 4, p. 50
40. P. Expert, T. S. Evans, V.D. Blondel, R. Lambiotte, Proc. Natl. Acad. Sci. USA **108**, 7663 (2011)
41. X. Liu, T. Murata, K. Wakita, Phys. Rev. E **90**, 012806 (2014)
42. M. MacMahon, D. Garlaschelli, Phys. Rev. X **5**, 021006 (2015)
43. P. Eades, X. Lin, W.F. Smyth, Inform. Process. Lett. **47**, 319 (1993)
44. M.E.J. Newman, M. Girvan, Phys. Rev. E **69**, 026113 (2004)
45. A. Arenas, J. Duch, A. Fernández, S. Gómez, New J. Phys. **9**, 176 (2007)
46. J. Goñi, B. Corominas-Murtra, R.V. Solé, C. Rodríguez-Caso, Phys. Rev. E **82**, 1 (2010)
47. F.D. Malliaros, M. Vazirgiannis, Phys. Rep. **533**, 95 (2013)
48. M.E.J. Newman, Proc. Natl. Acad. Sci. USA **103**, 8577 (2006)
49. M.E.J. Newman, Phys. Rev. E **74**, 036104 (2006)
50. U. Brandes, D. Delling, M. Gaertler, R. Görke, M. Hoefer, Z. Nikoloski, D. Wagner, IEEE Trans. Knowl. Data En. **20**, 172 (2008)
51. V.D. Blondel, J.-L. Guillaume, R. Lambiotte, E. Lefebvre, J. Stat. Mech.: Theory Exp. **2008**, P10008 (2008)
52. G. Csárdi and T. Nepusz, InterJournal Complex Systems, 1695 (2006)
53. J. Leskovec, J. Kleinberg, C. Faloutsos, in *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, Chicago, 2005* (ACM, New York, 2005), p. 177
54. J. Gehrke, P. Ginsparg, J. Kleinberg, SIGKDD Explor. **5**, 149 (2003)
55. <http://snap.stanford.edu/data/cit-HepPh.html> (Accessed: February 20, 2015)
56. U.N. Raghavan, R. Albert, S. Kumara, Phys. Rev. E **76**, 036106 (2007)
57. B.H. Good, Y.-A. de Montjoye, A. Clauset, Phys. Rev. E **81**, 046106 (2010)
58. S. Mangan, U. Alon, Proc. Natl. Acad. Sci. USA **100**, 11980 (2003)
59. <http://wvs.weizmann.ac.il/mcb/UriAlon/index.php?q=e-coli-transcription-network> (Accessed: February 17, 2015)
60. M.E.J. Newman, Phys. Rev. E **69**, 066133 (2004)

**Open Access** This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.